Consistency Analysis of Nearest Subspace Classifier

Yi (Grace) Wang,

Department of Mathematics, Duke University, Durham, NC, USA
yiwang@math.duke.edu

August 18, 2014

Abstract

The Nearest subspace classifier (NSS) finds an estimation of the underlying subspace within each class and assigns data points to the class that corresponds to its nearest subspace. This paper mainly studies how well NSS can be generalized to new samples. It is proved that NSS is strongly consistent under certain assumptions. For completeness, NSS is evaluated through experiments on various simulated and real data sets, in comparison with some other linear model based classifiers. It is also shown that NSS can obtain effective classification results and is very efficient, especially for large scale data sets. Nearest Subspace, Classification, Consistency, Unsupervised Learning

1 Introduction

The problem of classification is to construct a mapping that can correctly predict the classes of new objects, given training examples of old objects with ground truth labels [36]. It is a classical problem in statistical learning and machine learning and has been widely used in computer vision, pattern recognition, bioinformatics, etc. Examples of applications include face recognition, handwriting recognition and micro-array classification.

More precisely, this problem can be formalized as follows. Given a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, the goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$ such that $f(\mathbf{x})$ is a good approximation of y for the given $\mathbf{x}_i$'s as well as for new instances $\mathbf{x}$. Typically, $\mathcal{X}$ is a continuous domain and $\mathcal{Y}$ is a finite discrete set.

In the past few decades, a tremendous amount of work has been produced for this problem. Many approaches have been proposed, e.g., K-Nearest Neighbors (KNN) [21, 13, 18], Fisher's Linear Discriminant Analysis (LDA) [20, 42], Artificial Neural Networks (ANN) [43, 56, 35], Support Vector Machines (SVM) [7, 11, 44], and Decision Trees (see [8, 40, 41] for some well known algorithms). We refer to [23, 5] for a more careful overview of classification techniques.

Among this work is a class of methods based on subspace models. The compelling interest in subspace models can be attributed to their validation in real data. For instance, it has been justified that the set of all images of a Lambertian object (e.g., face images) under a variety of lighting conditions can be accurately approximated by a low-dimensional linear subspace (of dimension at most 9) [19, 24, 4]. Another example is that, under the affine camera model, the coordinate vectors of feature points from a moving rigid object lie in an affine subspace of dimension at most 3 (see [12]). These applications give rise to modeling data by subspaces; the study of subspace based classifiers is an important branch.

The first work in this category was CLAss Featuring Information Compression (CLAFIC) [55] (also known as Nearest SubSpace (NSS) classifier [39]; for the information contained in this name, we will adopt the usage of NSS throughout the paper). In this algorithm, each class is represented by a linear subspace and data instances are assigned to the nearest subspace. Instead of obtaining good representation of subspaces in NSS, the Learning Subspace Method (LSM) [28] proposes to learn the subspaces based on good discrimination (see [37] for more variants and discussions). The simple idea of subspace classifiers has been extended to nonlinear versions in various ways; many have shown state-of-the-art performance (see [48, 10, 33] for example and Section 2.3 for more details). After the first subspace analysis of face images [26, 51], classification approaches with subspace models have been used successfully in face recognition [9], handwritten digit recognition [29], speech recognition [27] as well as biological pattern recognition problems [38].

Although the design of subspace-based classification techniques has been actively explored, their theoretical justification is very under-studied. In this paper, we restrict our interests of justification to analyzing how well the classifiers can be generalized to new samples. By doing so, one can learn quantitatively how reliable the classification approaches are and can thus also guide the algorithm design accordingly. For this purpose, a functional (known as *risk function*) is used to measure the prediction quality of every classifier. More precisely, we assume X and Y being random variables; instances $\mathbf{x}_i$ and y_i are drawn independently from the distributions of X and Y respectively. For a classifier $f(x)$, its risk functional is defined as:

$$R(f) = \mathbb{E}_{(X,Y)} \mathbb{1}(f(X) \neq Y)$$

Based on this, the *optimal Bayes rule* is defined to be the classifier whose risk functional is minimal. The Bayes rule is optimal in the sense that its expected loss (defined as 1 when the predicted class is not equal to the truth) is minimal. Note that, since the actual distribution of (X, Y) is unknown, the Bayes rule is thus not available in reality. A natural desirable property of practical classifiers is having as small risk functional as possible. In this spirit, the property *consistency* is defined as the fact that the risk function converges to that of the optimal Bayes rule. In other words, classifier that is **not** consistent produces larger misclassification errors on average than the best scenario, no matter how

many data samples are available. Many classification algorithms, such as, KNN, SVM, LDA and some boosting methods [1, 47, 46, 53, 6, 3], have been shown to be consistent under certain conditions.

In this paper, we study the consistency property of the Nearest SubSpace (NSS) classifier. We prove its strong consistency under certain conditions. We also validate the performance of NSS through fruitful experiments, in comparison with other linear classifiers, LDA, FDA and SVM. These experiments demonstrate that NSS has very effective and comparable performance as its better known and more popular competitors. Since the classifiers under consideration are all simple and fundamental ones, they are not state-of-the-art. However, they are very important components of classification and such an experimental comparison completes the understanding of NSS. For our best knowledge, an experimental comparison like this (between NSS and other typical linear classifiers) has not been demonstrated yet. In the rest of the paper, we will begin with a description of the NSS algorithm (Section 2), followed by its consistency analysis (Section 3) and experiments (Section 4).

2 The NSS Algorithm and its Strong Consistency

For most of the applications, it suffices to assume that $\mathcal{X} \subset \mathcal{B}(0, M) \subset \mathbb{R}^D$ and $\mathcal{Y} = \{1, \cdots, K\}$, where $\mathcal{B}(0, M)$ is the ball centered at the origin with radius M and D and K are some positive integers. We will restrict ourselves to this case throughout the paper.

2.1 The NSS Algorithm

The NSS classifier assumes data lie on multiple affine subspaces, finds an estimate for these subspaces and assigns each instance to the nearest subspace. The following is a summary of the NSS algorithm.

Note that the closed form solution to (1) is the Singular Value Decomposition (SVD) of the centered data matrix for the kth class; such a data matrix consists of $\left((\mathbf{x}_{k_1} - \hat{\mathbf{u}}_k), \quad \cdots, \quad (\mathbf{x}_{k_{n_k}} - \hat{\mathbf{u}}_k) \right)$ with $\mathbf{x}_{k_1}, \cdots, \mathbf{x}_{k_{n_k}} \in C_k$.

2.2 The Main Theorem

As mentioned in Section 1, a desirable property for classifiers is *consistency*. Denote h_n to be any classification rule determined from n samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, f^* as the optimal Bayes rule, i.e., $f^* = \arg\min_f R(f)$ and $R^* := R(f^*)$ as its risk. Now we define strong consistency in the following sense.

Definition 1 (Strong Consistency). *A classification rule h_n is said to be strongly consistent if*

$$R(h_n) \to R^* \quad a.s.$$

3

Algorithm 1 Nearest Subspace (NSS) Classification

Require: $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ and d: intrinsic dimension, some positive integer and $d < D$.
Ensure: A function $f : \mathcal{X} \to \mathcal{Y}$.
  for $k = 1$ **to** K **do**

$$
\begin{aligned}
\hat{\mathbf{u}}_k &= \frac{1}{n_k} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i; \; C_k = \{\mathbf{x}_i : y_i = k\}; \; n_k = |C_k|. \\
\hat{\mathbf{B}}_k &= \underset{\substack{\mathbf{B} \in \mathbb{R}^{D \times d} \\ \mathbf{B}^T \mathbf{B} = \mathbf{I}_d}}{\arg\min} \sum_{\mathbf{x}_i \in C_k} \|(\mathbf{I} - \mathbf{B}\mathbf{B}^T)(\mathbf{x}_i - \hat{\mathbf{u}}_k)\|^2. \quad\quad (1)
\end{aligned}
$$

  end for
  $\hat{f}(\mathbf{x}) = \underset{k}{\arg\min} \|(\mathbf{I} - \hat{\mathbf{B}}_k \hat{\mathbf{B}}_k^T)(\mathbf{x} - \hat{\mathbf{u}}_k)\|^2.$

Since the NSS classifer is also based on n samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, from now on, we denote it as $\hat{f}_n$ for it for the rest of the paper. Then we obtain the following theorem for the NSS classifier described in Algorithm 1.

Theorem 1. *The NSS classifier $\hat{f}_n$ is strongly consistent, i.e., $R(\hat{f}_n) \to R^*$ a.s., when the following assumptions hold.*

(1) $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ are i.i.d. samples of random variable (X, Y); $X \in \mathbb{R}^D$ and $Y \in \{1, \dots, K\}$.

(2) $\mathbb{P}(Y = i) = \frac{1}{K}$.

(3) $X | Y = k \sim \mu_{L_k} \times \mu_{L_k^\perp}$; L_k is the underlying d-dimensional subspace for the kth class; μ_{L_k} is a uniform measure on $L_k \cap \mathcal{B}(\mathbf{u}_k, M)$ (a bounded ball centered at $\mathbf{u}_k$, the underlying center for the kth class); $\mu_{L_k^\perp}$ is a measure on $L_k^\perp$ decreasing exponentially w.r.t. the square distance from L_k;

This theorem reveals that the average prediction error of NSS converges to the optimal prediction error under certain conditions. It is a similar but slightly weaker result in contrast to that for LDA in [53], since the above condition (3) is stronger than that for LDA. Note that both results are about consistency for a class of distributions. On the other hand, the consistency results for KNN, SVM and some boosting methods are for all distributions, and thus are more general [47, 46, 6, 3].

2.3 Discussions

The NSS algorithm is a very simple and basic classification method, since it assumes linear structure in data. Linear models have their limitations, since the linearity constraint often is not satisfied in real data. However, they are important for the following reasons: (1) Linear classifiers are easy to compute and analyze. (2) They are a first order approximation for the true classifier. (3)

They often have good interpretations, critical in many applications. (4) Linear models are the best that can be done when the available training data are limited. (5) Linear models are the foundation from which more complex models can be generalized (see [23] for more discussion). Therefore, it is important to study this class of methods thoroughly, even if in practice they are no longer state-of-the-art. The computational complexity and the extensions of the NSS algorithm are further discussed as follows.

Complexity. It is worth pointing out that the NSS algorithm is efficient. Assuming $D < n$, the computational complexity of the training process of NSS is $\mathcal{O}(KD^2(\bar{n}_k + 2D))$ where $\bar{n}_k = \frac{1}{K}n_k$. LDA and FDA have similar complexity since they all require some eigen- or singular value decomposition operations. On the other hand, SVM requires $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ operations. Therefore, for large scale (n large and $n \gg D$) problems, the computation of NSS is much faster than for linear SVM. In cases when the data is of large scale and some sensible results are needed quickly, NSS is a good choice. Section 4 will provide more details of the performance in terms of both accuracy and speed of the algorithms.

Extensions. The NSS method has been modified and extended through different methods: localization, the kernel trick and the hybrid model. The local subspace methods find, for the investigated data sample, their nearest neighbors in each class and attribute by their distances to the subspace spanned by these neighbors [45, 29, 54, 10, 33]. Due to the fact that only an inner product is needed in the NSS algorithm, it can be naturally extended by the kernel trick, where the original data are embedded into a higher dimensional space and subspace structures are learned there [29, 50, 2, 57, 34]; these two techniques are combined in [58]. Another direction is to represent each class by multiple subspaces [29, 48, 33], where [48] also uses a more general metric than the Euclidian distance. All of these extended techniques define nonlinear decision boundaries and the recent works [48, 10, 33] have shown their state-of-the-art performance.

3 Proof of Theorem 1

In this section, we give a complete proof of Theorem 1 following [53].

3.1 Notations

We first describe the problem in detail and prepare to prove the theorem. Consider a classification problem, where the goal is to assign an individual instance to one of K classes, given n observations of (X, Y). To do this, the space $\mathbb{R}^D$ is partitioned into subsets $H_1, \ldots, H_K$ such that, for $k = 1, \ldots, K$, the individual instance is classified to be in group k when $X \in H_k$. This procedure generates a discriminant rule as a mapping $f : \mathbb{R}^D \to \{1, \ldots, K\}$ that takes the value $f(X) = k$ whenever the individual is assigned to the kth group, and this can be

written as $f(X) = \sum_{k=1}^{K} k \mathbb{1}_{H_k}(X)$, where $\mathbb{1}_{H_k}(X)$ is the indicator function of the subset H_k.

Let Y be the discrete random variable (class index or group label) which represents the true membership of the individual under study. Denote the class prior probabilities $\pi_k = \mathbb{P}[Y = k] > 0$, $\sum_{k=1}^{K} \pi_k = 1$ and $k = 1, \ldots, K$. Furthermore, assume there exist density functions $g_k(X)$ such that $\mathbb{P}[X \in \mathcal{A}|Y = k] = \int_{\mathcal{A}} g_k(X)dX$, $k = 1, \ldots, K$ for $\mathcal{A}$, a subset of $\mathbb{R}^D$.

Given (X, Y), the rule $f(X) = \sum_{k=1}^{K} k I_{H_k}(X)$ is in error when $f(X) \neq Y$ and its probability of misclassification is computed as:

$$
\begin{aligned}
R(f) &= \mathbb{E}_{(X,Y)} \mathbb{1}(f(X) \neq Y) = \mathbb{P}[f(X) \neq Y] = 1 - \mathbb{P}[f(X) = Y] \\
&= 1 - \sum_{k=1}^{K} \mathbb{P}[X \in H_k, Y = k] \\
&= 1 - \sum_{k=1}^{K} \mathbb{P}[Y = k]\mathbb{P}[X \in H_k|Y = k] \\
&= 1 - \sum_{k=1}^{K} \pi_k \int_{H_k} g_k(X)dX.
\end{aligned}
\tag{2}
$$

The rule $f^* = \sum_{k=1}^{K} k \mathbb{1}_{H_k^*}(X)$ that minimizes (2), or the Bayes rule, is given by the partition

$$
H_k^* = [X : \pi_k g_k(X) = \max_{1 \leq j \leq K} \pi_j g_j(X)], \quad k = 1, \ldots, K.
$$

Then the corresponding optimal error is:

$$
R^* = R[f^*(X)] = 1 - \sum_{k=1}^{K} \pi_k \int_{H_k^*} g_k(X)dX.
$$

In general, both π_k and g_k are unknown, so rules used in practice are sample based rules of the form $\hat{f}_n(X) = \sum_{k=1}^{K} k I_{\hat{H}_{k,n}}(X)$, where the subsets $\hat{H}_{k,n}$ depend on the data set $\Omega_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ formed by n i.i.d. observations from (X, Y). The appropriate measure of error of a sample rule $\hat{f}_n(X)$ is $R_n = \mathbb{P}[\hat{f}_n(X) \neq Y]$.

3.2 Proof of Theorem 1

We will first prove a useful lemma which gives a bound for $R_n - R^*$.

Lemma 1. *Assume $\pi_k = \frac{1}{K}$ and let $\hat{g}_{k,n}(X)$ be an estimate of $g_k(X)$ from Ω_n , for $k = 1, \cdots, K$. Let $\hat{f}_n(X)$ be the classifier derived from $\hat{g}_{k,n}(X)$, i.e., $\hat{f}_n(X) = \arg\max_k \hat{g}_{k,n}(X)$. Then*

$$
0 \leq R_n - R^* \leq \frac{1}{K} \sum_{k=1}^{K} \int |g_k(X) - \hat{g}_{k,n}(X)|dX.
$$

6

Proof. Since $\pi_k = \frac{1}{K}$, we have $H_k^* = [X : g_k(X) = \max_{1 \le j \le K} g_j(X)$. Thus,

$$
\begin{aligned}
R^* &= 1 - \frac{1}{K} \sum_{k=1}^{K} \int_{H_k^*} g_k(X) dX = 1 - \frac{1}{K} \int \max_k g_k(X) dX \\
&\le 1 - \frac{1}{K} \int g_{\hat{f}_n}(X) dX = R_n
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
R_n - R^* &= \frac{1}{K} \int (\max_k g_k(X) - g_{\hat{f}_n}(X)) dX \\
&= \frac{1}{K} \int (\max_k g_k(X) - \hat{g}_{\hat{f}_n,n}(X)) dX + \frac{1}{K} \int (\hat{g}_{\hat{f}_n,n}(X) - g_{\hat{f}_n,n}(X)) dX \\
&= \frac{1}{K} \int (\max_k g_k(X) - \max_k \hat{g}_{k,n}(X)) dX + \frac{1}{K} \int (\hat{g}_{\hat{f}_n,n}(X) - g_{\hat{f}_n,n}(X)) dX \\
&\le \frac{1}{K} \sum_{k=1}^{K} \int |g_k(X) - \hat{g}_{k,n}(X)| dX.
\end{aligned}
$$

$\square$

A similar result of Lemma 1 can be found in the Theorem 1 in [16] (p. 254). Now we prove the main theorem of our paper.

Proof of Theorem 1. Due to condition (2), we have

$$
H_k^* = [X : g_k(X) = \max_{1 \le j \le K} g_j(X)].
$$

On the other hand, based on the assumption (3), the density functions can be written as

$$
\begin{aligned}
g_k(X) &= C(d)\beta \exp{(-\alpha t)}, \\
t &= (X - \mathbf{u}_k)^T (I - \mathbf{P}_k)(X - \mathbf{u}_k)
\end{aligned}
$$

for some $\alpha > 0$, β independent of t, constant $C(d)$ and $\mathbf{P}_k = \mathbf{B}_k \mathbf{B}_k^T$ with $\mathbf{B}_k$ being the orthonormal basis for L_k.

Then the classifier generated by the Algorithm 1 can be written as:

$$
\hat{f}_n(X) = \sum_{k=1}^{K} k I_{\hat{H}_{k,n}}
$$

with the following notation:

$$
\begin{aligned}
\hat{\mathbf{P}}_k &= \hat{\mathbf{B}}_k \hat{\mathbf{B}}_k^T \\
\hat{g}_{k,n}(x) &= C(d)\beta \exp{(-\alpha(X - \hat{\mathbf{u}}_k)^T (I - \hat{\mathbf{P}}_k)(X - \hat{\mathbf{u}}_k))} \\
\hat{H}_{k,n} &= [X : \hat{g}_{k,n}(X) = \max_{1 \le j \le K} \hat{g}_{j,n}(X)]
\end{aligned}
$$

7

Thus the NSS classifier can be considered as a plug-in version of the Bayes rule. By Lemma 1, the difference $R_n - R^*$ can be bounded in the form

$$0 \leq R_n - R^* \leq \frac{1}{K} \sum_{k=1}^{K} \int_{\mathbb{R}^D} |g_k(X) - \hat{g}_{k,n}(X)| dX$$

For each fixed $1 \leq k \leq K$, we have

$$0 \leq \int_{\mathbb{R}^D} |g_k(X) - \hat{g}_{k,n}(X)| dX \leq \int_{\mathbb{R}^D} g_k(X) + \hat{g}_{k,n}(X) dX < \infty$$

Therefore, it suffices to show that $\hat{g}_{k,n} \to \hat{g}_k$ $a.s$ and due to the continuity of $g(\cdot)$, to show $\hat{\mathbf{u}}_k \to \mathbf{u}_k$ and $\hat{\mathbf{P}}_k \to \mathbf{P}_k$ $a.s.$ The fact that $\hat{\mathbf{u}}_k$ and $\hat{\mathbf{P}}_k$ are the maximum-likelihood estimations (MLE) of $\mathbf{u}_k$ and $\mathbf{P}_k$ completes the proof. $\quad\square$

4   Experiments

In this section, we evaluate the performance of the NSS algorithm through various experiments and compare it with LDA, FDA and linear SVM. The purpose of demonstrating these results is two-fold. First, it is to show that, as a simple and basic method, the NSS algorithm can obtain very useful results and is comparable to its competitors. Second, it serves as a complementary perspective to the theoretical portion of the paper. The reason why we include LDA, FDA and linear SVM in the comparison is because they are similar to NSS. Note that the objective is not to prove NSS is state-of-the-art. On the other hand, the significance of studying this method has been fully discussed in Section 2.3.

4.1   Data

We test the classification methods on two simulated data sets and five real data sets. In the following, we will give a brief description for each of them and a summary of size, dimension and the number of classes can be found in Table 1.

Mixture Gaussian.   Data samples are generated from $K = 3$ Gaussian distributions in $\mathbb{R}^3$ with means $\mu_1 = (1, 2, 3)^T$, $\mu_2 = (-1, -2, -3)^T$, $\mu_3 = (-1, 2, -3)^T$ and variances $\Sigma_1 = \begin{pmatrix} 3 & 0.2 & 0.1 \\ 0.2 & 2 & 0.2 \\ 0.1 & 0.2 & 2 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\Sigma_3 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$.
The total number of samples is 1200; 400 in each class.

Multiple Subspaces.   For the multiple subspaces experiment, data are generated uniformly from 3 2-dimensional linear subspaces (bounded in a unit disk) in $\mathbb{R}^{50}$. The angles between the subspaces are at least $\frac{\pi}{8}$. Gaussian noise with 0 mean and 0.05 standard deviation is added. Again, 1200 samples are generated in total with 400 in each class.

Wine. Wine recognition data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The goal is to determine the types of wines from the quantities of 13 constituents found in them. The data were first collected in [22] and now can be found in the UCI machine learning repository.

DNA. We use the Statlog version of the primate splice-junction DNA data set (found in [49]). The problem is to recognize, given a sequence DNA, the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out). The features are binary variables representing nucleotides in the DNA sequence. Three classes are "intron to exon" boundary, "exon to intron" boundary and neither. This data set has three subsets, training, evaluation and testing. All of them are used in our experiments.

USPS. USPS [15] is a database of scanning images of handwritten digits from US Postal Services envelopes. The goal is to recognize digits given their 16×16 grayscale images. Both the training and testing sets are used in our experiments.

Vehicle. This Vehicle data set [17] collects signals obtained by both acoustic and seismic sensors and the goal is to classify vehicle types from the original data. It has two subsets, training and testing, and we use both of them in our experiments.

News20. The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The problem imposed here is to recognize the newsgroups from texts. This data was originally collected in [30]. Due to the very large scale of the data, we use only the testing set in our experiment for simplicity.

Table 1: A summary of the data sets

data	data size (# of samples)	# of classes	ambient dimension (# of features)	reduced dimension
Mixture Gaussian	1200	3	3	NA
Multiple Subspaces	1200	3	50	NA
Wine	178	3	13	NA
Vehicle	98,528	3	100	NA
DNA	3,186	3	180	NA
USPS	9,298	10	256	38
News20	3,993	20	62,060	1000

4.2 Implementation Details

Real data used in our experiments are originally from the UCI machine learning repository, Statlog and other collections. We download them from [49], where data samples have been scaled linearly to be within [0, 1] or [-1, 1]. For the data sets USPS and News20, the ambient dimension is reduced by Principal Component Analysis to be at most 1000 and such that 95% variance is explained. The reduced dimension for the USPS and News20 data sets is shown in the last column of Table 1. For NSS, the intrinsic dimension of the subspaces is determined by 10-fold cross validation.

The classification experiments are carried out in Matlab. We use the default function *classify* of the Statistics toolbox for LDA. For multiclass FDA and SVM (see [32] and [14]), we use implementations from [25] and [31]. The NSS can be simply realized and the version we use can be found from the author's homepage http://www.math.duke.edu/~yiwang/.

4.3 Results

For each data set, we randomly split it into two subsets, each with 80% and 20% of the data, and use the former as the training set and the latter as the testing set. All experiments are repeated 200 times for the simulated data sets and 10 times for the real data sets, including the random generation (for the simulated data sets) and the random splitting processes. The mean and standard deviation of the accuracy of all methods under investigation are reported in Table 2, while the running time for the training process is recorded in Table 3.

Table 2: A summary of classification results: mean accuracy $\pm$ standard deviation (%)

Data	Methods			
	NSS	LDA	FDA	SVM
Gaussian	88.11 ± 4.47	$\mathbf{95.12} \pm 1.58$	81.43 ± 3.74	94.93 ± 1.58
Subspace	$\mathbf{99.16} \pm 0.51$	34.97 ± 2.73	33.74 ± 2.26	46.57 ± 3.23
Wine	94.29 ± 3.81	$\mathbf{98.57} \pm 2.02$	92.57 ± 3.86	96.29 ± 1.93
Vehicle	74.23 ± 0.34	$\mathbf{80.15} \pm 0.22$	75.11 ± 0.33	NA
DNA	90.28 ± 1.05	$\mathbf{93.23} \pm 1.02$	78.59 ± 2.07	91.11 ± 0.94
USPS	$\mathbf{96.54} \pm 0.39$	91.19 ± 0.58	48.58 ± 1.01	94.02 ± 0.48
News20	75.18 ± 1.60	35.55 ± 1.72	9.50 ± 1.39	$\mathbf{75.54} \pm 1.26$

From the above results, we know that the NSS algorithm can obtain results comparable to its better known competitors LDA and SVM, for a broad range of classification problems. Meanwhile, the computation is very fast, roughly the same order as FDA and LDA, but significantly faster than SVM, especially for large scale problems. Additionally, LDA requires that the covariance matrix is positive definite, which is not satisfied in some high dimensional data sets. This is another reason why we reduce the ambient dimension for the USPS and News20 datasets. However, NSS does not have this restriction.

Table 3: A summary of running time on real data sets (seconds)

Data	Methods			
	NSS	LDA	FDA	SVM
Wine	8.175×10^{-4}	0.012	0.019	0.654
Vehicle	1.064	1.206	3.449	NA
DNA	0.042	0.037	0.212	73.685
USPS	0.035	0.052	0.148	935.296
News20	1.089	1.018	12.841	168.209

5 Conclusion

In this paper, we reviewed a simple classification algorithm (NSS) based on the model of multiple subspaces. We proved its strong consistency under certain conditions, which means that under these conditions, the prediction error of NSS on average converges to that of the optimal classifier. Finally, we evaluated NSS on various data sets and compared it with its competitors. Results showed that NSS can obtain very useful results efficiently, especially for large scale data sets.

By studying the consistency property of NSS, we are inspired to further explore subspace-based classification methods along the following directions in the future. First, NSS finds a good estimation for the underlying subspace models by minimizing the sum of squares of fitting errors. However, for the purpose of classification, it is more helpful to obtain models which can "separate" or "discriminate" classes. Therefore, in order to improve the classification performance, some separation measure can be taken into account. In fact, an advanced supervised learning method based on multiple subspaces has been proposed [48]. It would be fruitful to analyze this method or other variants theoretically.

Moreover, a general way to find a good classifier is to minimize an empirical risk function, which is typically defined as $R_{\mathrm{emp}}(f) = \sum_{i=1}^{n} \mathbb{1}(f(\mathbf{x}_i) \neq y_i)$. This idea can be combined with the multiple subspaces model. Similar approaches to that in [52] can be applied to analyze its consistency.

Acknowledgements

References

[1] Francis Bach and Jean-Yves Audibert. Supervised learning for computer vision: Theory and algorithms, 2008.

[2] T. Balachander and R Kothari. Kernel based subspace pattern classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1999.

[3] Peter L. Bartlett and Mikhail Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.

[4] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[6] Gilles Blanchard, G´bor Lugosi, and Nicolas Vayatis. On the rate of convergence of regularized boosting classifiers. *J. Mach. Learn. Res.*, 4:861–894, December 2003.

[7] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[9] Raffaele Cappelli, Dario Maio, and Davide Maltoni. Multispace kl for pattern representation and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):977–996, September 2001.

[10] Hakan Cevikalp, Diane Larlus, Marian Neamtu, Bill Triggs, and Frdric Jurie. Manifold based local classifiers: Linear and nonlinear approaches. *Signal Processing Systems*, 61(1):61–73, 2010.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

[12] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[13] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, September 2006.

[14] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, March 2002.

[15] Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann, 1990.

[16] L. Devroye and L. Györfi. *Nonparametric density estimation: the L_1 view.* John Wiley, New York, 1985.

[17] Marco F. Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *J. Parallel Distrib. Comput.*, 64(7):826–838, July 2004.

[18] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, NY, 1973.

[19] R. Epstein, P.W. Hallinan, and A.L. Yuille. 5 ± 2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. In *Physics-Based Modeling in Computer Vision, 1995., Proceedings of the Workshop on*, page 108, jun. 1995.

[20] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[21] Evelyn Fix and Jr. Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. Technical Report Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolf Field, Texas, 1951.

[22] M. Forina et al. Parvus - an extendible package for data exploration, classification and correlation. Institute of Pharmaceutical and Food Analysis and Technologies. Via Brigata Salerno, 16147 Genoa, Italy, 1991.

[23] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations.* New York: Springer-Verlag, 2001.

[24] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 11–18, 2003.

[25] Darko Juric. Multiclass lda. `http://www.mathworks.com/matlabcentral/fileexchange/31760-multic`, 2011.

[26] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, January 1990.

[27] T. Kohonen, H. Riittinen, M. Jalanko, E. Reuhkala, and S. Haltsonen. A thousand-word recognition system based on the learning subspace method and redundant hash addressing. In *Proceedings of the 5th Intertional Conference on Pattern Recognition*, pages 158–165, Dec. 1-4 1980.

13

[28] Teuvo Kohonen, Gábor Németh, Kalle-J. Bry, Matti Jalanko, and Heikki Riittinen. Spectral classification of phonemes by learning subspaces. In *ICASSP*, pages 97–100, 1979.

[29] Jorma Laaksonen. *Subspace Classifiers in Recognition of Handwritten Digits*. PhD thesis, Helsinki University of Technology, 1997.

[30] Ken Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*, 1995.

[31] F. Lauer and Y. Guermeur. MSVMpack: a multi-class support vector machine package. *Journal of Machine Learning Research*, 12:2269–2272, 2011. `http://www.loria.fr/~lauer/MSVMpack`.

[32] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and Information Systems*, 10(4):453–472, 2006.

[33] Yiguang Liu, Shuzhi Sam Ge, Chunguang Li, and Zhisheng You. k-ns: A classifier by the distance to the nearest subspace. *IEEE Transactions on Neural Networks*, 22(8):1256–1268, 2011.

[34] Eisaku Maeda and Hiroshi Murase. Kernel-based nonlinear subspace method for pattern recognition. *Systems and Computers in Japan*, 33(1):38–52, 2002.

[35] Marvin Lee Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge Mass., expanded ed. edition, 1988.

[36] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[37] E. Oja. *Subspace methods of pattern recognition*. Electronic & electrical engineering research studies. Research Studies Press, 1983.

[38] Oleg Okun. Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm. In *Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics*, 2004.

[39] Fatih Porikli and Yuejie Chi. Connecting the dots in multi-class classification: From nearest subspace to collaborative representation. *IEEE Conference on Computer Vision and Pattern Recognition*, 0:3602–3609, 2012.

[40] J. Ross Quinlan. Discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Micro-Electronic Age*, pages 168–201. Edinburgh University Press, Edinburgh, 1979.

[41] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[42] C.R. Rao. *Advanced statistical methods in biometric research*. Wiley publications in statistics. Wiley, 1952.

[43] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[44] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *First International Conference on Knowledge Discovery and Data Mining*, Menlo Park, 1995. AAAI Press.

[45] Wladyslaw Skarbek, Miloud Ghuwar, and Krystian Ignasiak. Local subspace method for pattern recognition. In Gerald Sommer, Konstantinos Daniilidis, and Josef Pauli, editors, *CAIP*, volume 1296 of *Lecture Notes in Computer Science*, pages 527–534. Springer, 1997.

[46] Ingo Steinwart. Support vector machines are universally consistent. *J. Complexity*, 18(3):768–791, 2002.

[47] Charles J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–620, 1977.

[48] Arthur Szlam and Guillermo Sapiro. Discriminative k-metrics. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 1009–1016, Montreal, June 2009. Omnipress.

[49] G. Towell, M. Noordewier, and J. Shavlik. Libsvm data: Classification (multi-class). http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html, 1992.

[50] Koji Tsuda. Subspace classifier in reproducing kernel hilbert space. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1999.

[51] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, January 1991.

[52] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[53] Santiago Velilla and Adolfo Hernández. On the consistency properties of linear and quadratic discriminant analyses. *Journal of Multivariate Analysis*, 96(2):219–236, October 2005.

[54] Pascal Vincent and Yoshua Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 985–992. MIT Press, 2001.

[55] S. Watanabe, P. F. Lambert, C. A. Kulikowski, J.L. Buxton, and R. Walker. Evaluation and selection of variables in pattern recognition. In J. Tou, editor, *Computer and information sciences*, volume 2, pages 91–122. Academic Press, New York, 1967.

[56] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

[57] W. Zhao. Subspace methods in object/face recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1999.

[58] Dongfang Zou. Local subspace classifier in reproducing kernel hilbert space. In Tieniu Tan, Yuanchun Shi, and Wen Gao, editors, *ICMI*, volume 1948 of *Lecture Notes in Computer Science*, pages 434–441. Springer, 2000.